

REMARKS

This application was filed on 05 October 2001 with 58 claims. In the first examination of the application, the Examiner, in correspondence dated 17 December 2003, objected to the Drawings, the Specification, and claim 10 because of a typographical error. The Examiner further rejected claims 1-58 under 35 U.S.C. §103(a) over U.S. Patent No. 6,088,694 entitled **CONTINUOUS AVAILABILITY AND EFFICIENT BACKUP FOR EXTERNALLY REFERENCED OBJECTS** to Burns et al. (Burns '694) in view of U.S. Patent No. 6,581,075 entitled **SYSTEM AND METHOD FOR DATABASE SYNCHRONIZATION** to Guturu et al. (Guturu '075). In response, Applicant amends the abstract to remove the title of the application and amends claim 10; by doing so, has overcome the informal objections. Applicant respectfully requests to be allowed to submit corrections to the drawings when Attorney for Applicant receives copies of the drawings that were submitted and reviewed by the Examiner and the Official Draftsman. Applicant further amends the independent claims to include limitations of some of the dependent claims, and cancels some dependent claims. Claims 1-11, 14-30, 33-48, 51-58 are pending.

The Rejection of claims 1-58 under 35 U.S.C. §103(a)

The Examiner rejected claims 1-58 under 35 U.S.C. §103(a) over Burns '694 in view Guturu '075. In response, Applicant amends the claims to distinctly point out and particularly claim that the "object can be edited independently of the related metadata in a loose transaction model."

In amending the claims, Applicant does not add new matter. Support for the **loose transaction model** in the originally-filed application is set forth in the preamble of independent claim 1 and again in the originally-filed specification on page 9, lines 10-12, wherein a loose transaction model is one where the file can be edited independently of the metadata." }

Applicant further amend other claims to further point out and distinctly claim that the handle includes a filename that has an encrypted access token with a hash value, wherein the hash value includes the embedded version number. Again, with the amendment, Applicant does not add new matter. A handle is described in the originally-filed specification as "obtained from the mediator, that is preferably a

filename with an encrypted access token string embedded as part of the filename, and is supplied as the filename to the filesystem API." (page 11, lines 23-25 - see paragraph beginning with "The file is updated using normal file ....." ] Description of the encrypted access token string is further set forth on page 16, lines 3-8 which state:

For cases where consistency is to be maintained between the DBMS meta-data [sic] and the object, the token can be setup to contain a version number and a different one-way hash value, H, computed as:

H = hash (K, Tx, file name, server name, token length, flags, Vc), where Vc is the latest committed version at which the token is created and flags can contain information indicating that consistency is requested.

The Examiner alleges that Burns '694 discloses the elements of the claimed invention, i.e., "a method of maintaining consistency of content of an object and metadata related," "storing said related meta-data and a reference to said object in a table," "the object being stored externally to said database," "said reference used to obtain a handle for directly accessing or manipulating said external object", and "obtaining a version number." The Examiner, however, admits that Burns does not explicitly disclose the steps of comparing the embedded version number with a version number of a latest committed version of an externally stored object to determine if the handle refers to a current version of the externally stored object.

Respectfully, there is one major difference between Burns '694 and Applicant's claimed invention. Burns '694 provides for "transaction atomicity" (*Burns '694 at column 10, lines 65-66*) when modifying the object, i.e., the appends and updates of the file/object are conducted in a "transactional manner" (*Burns '694 at column 12, lines 3-8 and column 13, lines 8-13*). This "transactional manner" is distinguished from the **loose transaction model** claimed by Applicant in the amended claims.

In fact, Burns '694 is described by Applicant on page 1, lines 29-32 wherein the "file and meta-data updates are tightly coupled (i.e. both updates happen within a single unified transaction), a transaction coordinator typically ensures a consistent view by locking out readers of meta-data as well as file data until the transaction is committed. Intermediate / uncommitted updates to either are not visible." Burns '694 simply provides the other users with a "read-only" copy while the object is being updated or appended, thus preserving the "transactional manner." See Burns '694 at

column 9, lines 8-18 (Such constraints include, for example, making a database system the owner of the named file and marking the file as a read-only file. This linkage is provided in a transactional manner. The rationale for changing the owner of the file to the database system from a file system user is to prevent the file from being renamed or deleted by file system users, which guarantees the integrity of any reference made in the database system to the file. Marking the file as read only guarantees the integrity of indexes that may be created on the file and stored in the database system for search.”), column 4, lines 40-43 (“During this modification, other applications or users can access the immediately previous versions of the file which is registered in the DBMS.”) and column 4, lines 63-67 (“While an application or user is appending data, the file is still available to other applications or users through the Datalink column supported by the DBMS. Other applications accessing the file being appended only view the data that existed before the append function was invoked.”) Thus, given the teaching and methodology of Burns ‘694, *see Burns ‘694 at column 5, lines 25-40*, the object and its metadata are updated as part of the same transaction in a “transactional manner.”

A “transactional manner” to preserve data integrity, as in Burns ‘694, has a clear definition in the art of database management with the following characteristics: a “transactional manner” is **ATOMIC** meaning that either all or none of the operations are completed; is **CONSISTENT** meaning that all transactions must leave the database in consistent state; is **ISOLATED** meaning that the transactions can’t interfere with each other’s work and incomplete work isn’t visible to other transactions; and is **DURABLE** meaning that successful transactions must persist through crashes of the object/file. Quite simply, because of this “transactional manner”, Burns ‘694 cannot suggest or teach a **loose transaction model** in which the object/file can be modified independently of the metadata.

There are other differences too between Burns ‘694 and the claimed invention: Burns ‘694 and Applicant are solving different problems in a similar database system. Burns ‘694 solves the problem of permitting multiple users to access the object while it is being updated or appended; whereas Applicant solves the problem of how to maintain consistency between the object and its metadata as it is being updated or appended by multiple users. Another difference correctly noted by the Examiner is his observation that Burns ‘694 does not refer to an **encrypted access token**; the

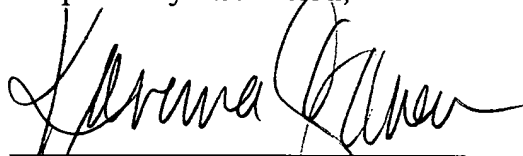
token **having a hash value**; the hash value **including a version number**. There is simply no mention of how Burns '694 ascertains which is the most current version of an object to be appended or updated; it is just assumed that the system has unrestricted access to both the object and the related metadata.

The Examiner then proposes that the version number, timestamp, operational priority, and node priority parameters as taught by Guturu '075 provide sufficient teaching and motivation to modify Burns '694. Applicant respectfully traverses because one of skill in the art would not look to Guturu '075, nor Burns '694 for that matter, because [Guturu '075 and Burns '694 are solving different problems than Applicant.] Applicant is solving the noted problem of maintaining consistency between file content and the associated metadata, whereas Guturu '075 solves the problem of synchronizing identical and redundant databases. See Guturu '075 at column 1, lines 11-18. Because of the task of maintaining redundant databases having identical data that is being updated simultaneously by multiple users, Guturu '075 assumes a "tightly coupled" or a "transactional manner" in that the metadata related to the object being updated is never separated from the object. In fact, absent Applicant's teaching, one of skill in the art would not look to Guturu '075 which maintains identical data in redundant databases by comparing versions or timestamps when maintaining related metadata with its object. Guturu '075 updates the object and the metadata at once and does not suggest otherwise. Respectfully, comparing version numbers, timestamps and priorities does not create a loose transaction model to maintain consistency between the metadata and the object itself, as claimed by Applicant.

Conclusion

Applicant amends the claims to particularly point out and distinctly claim a loose transaction model which is neither taught or suggested by Burns '694 or by Guturu '075 and to further refine the nature of the handle. Applicant requests the Examiner to review the amended claims and to pass the case to issuance. If the Examiner thinks it would expedite the prosecution and the issuance of the patent, he is encouraged to telephone the Attorney listed below.

Respectfully submitted,



By

Karuna Ojanen  
Registration No. 32,484  
(507) 285-9003 voice  
(507) 252-5345 fax

Date: 17 March 2004

IBM Corporation  
Intellectual Property Law, Department IQ0A  
Building 040-3, Office B009  
Endicott, New York 13760-5553